

A QUANTUM PERFORMANCE SIMULATOR BASED ON FIDELITY AND FAULT-PATH COUNTING

Andre Van Rynbach, Jeffrey Hussmann*, Jungsang Kim

*Department of Electrical and Computer Engineering, Duke University
Durham, NC 27705, USA*

Abhijit C. Mehta

*Department of Physics, Duke University
Durham, NC 27705, USA*

Ahsan Muhammad

*Department of Computer Science, Duke University
Durham, NC 27705, USA*

Quantum performance simulators can provide practical metrics for the effectiveness of executing theoretical quantum information processing protocols on physical hardware. In this work we present a scheme to simulate the performance of fault tolerant quantum computation by automating the tracking of common fault paths for error propagation through a circuit and quantifying the fidelity of each qubit throughout the computation. Our simulation tool outputs the expected execution time, required number of qubits and the final error rate of running common fault tolerant protocols on a universal hardware, assumed to be a network of qubits with full connectivity. Our technique efficiently estimates the upper bound of error probability and provides a useful performance measure of the error threshold at low error rates where conventional Monte Carlo methods are ineffective. To verify the proposed simulator, we present simulation results comparing the execution of quantum adders which constitute a major part of Shor's algorithm.

Keywords: Quantum Performance Simulator, fidelity, quantum adder circuit, resource tracking

1. Introduction

Quantum computation (QC) represents a new paradigm in computing capable of solving several classically intractable problems in polynomial time, and has been an intense field of research from both a physical hardware and computer science perspective over the past two decades. Various quantum algorithms have been discovered to show the power of QC, such as Shor's factoring algorithm [1] and Grover's unstructured search algorithm [2]. Quantum

*Present Address: Institute for Computational Engineering and Sciences, University of Texas at Austin

assembly languages have been developed to formalize the process of compiling algorithms into basic executable logic elements [3].

Just as with classical computers, effective mapping and optimized scheduling of the logic elements on practical hardware platforms requires knowledge of the hardware architecture. QC poses additional challenges on this mapping and scheduling process arising when the physical systems representing quantum bits (qubits) suffer from decoherence processes and faulty gate operations, leading to a loss of quantum information during the computation. Fault-tolerant (FT) procedures based on quantum error-correcting codes (QECCs) were established to overcome this challenge, which comes at the cost of additional physical resources, computation time and higher levels of operational complexity. The *threshold theorem* proves that scalable QC is possible using error correcting codes, as long as (1) the error probability for the physical process is below a certain threshold value (*error threshold*), and (2) sufficient quantum and classical resources are available to implement the error correction codes necessary to ensure FT [4]. Although detailed procedures for achieving FT are well-established, quantitative estimates of the resource overhead, computation time and resulting performance gain (in terms of reduced error rate) require an automated software tool due to the complexity in these procedures. Previous work has proven the existence of a rigorous lower bound on the error threshold for distance 3 concatenated codes, such as the Steane code considered here [5, 6]. Our work serves as a generalization of this type of analysis by numerically finding an error threshold and tracking the individual error rates on qubits throughout the execution of a full quantum circuit. This method overcomes the limitation of many Monte Carlo based analyses where simulating low error rates becomes prohibitively slow [13]. Furthermore, our quantum performance simulator keeps accurate track of the resource and computation time overhead of running a given circuit and, in addition to tracking the error probabilities, provides a means to evaluate the effectiveness of the hardware architecture and optimize the choice of FT strategy for a finite-size quantum processor.

2. Quantum Error Correcting Codes and Fault Tolerance

QECCs are needed to protect the qubits from the faulty physical processes used to prepare, store, and process the qubits, and form the basis of FT quantum computation (FTQC). In general, a QECC uses m physical qubits to encode k logical qubits in a way that some set of possible error events can be reliably corrected. The quantum error correction condition gives a necessary and sufficient condition for the code space $C = \{|\psi_i\rangle\}$ to correct a set of errors $E = \{E_a\}$: an error is correctible if and only if

$$\langle\psi_i|E_a^\dagger E_b|\psi_j\rangle = C_{ab}\delta_{ij} \quad (1)$$

for any $E_a, E_b \in E$ and any $|\psi_i\rangle, |\psi_j\rangle \in C$ for a set of real numbers C_{ab} [7]. Intuitively, this reflects the fact that the results of errors acting on codewords must be orthogonal in order to reliably distinguish between different possible errors, and that no information must be gained about the encoded qubit state in order to preserve quantum coherence. Typical errors on individual qubits cause arbitrary relative phase accumulation on a quantum state, but can be treated as discrete bit flip (X) or phase flip (Z) errors occurring with small probability. This is because the error states collapse into a definitive qubit state upon syndrome measurement when the errors are discretized [8]. A typical codeword for QECC is a highly entangled state

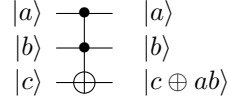


Fig. 1. Toffoli gate where \oplus is bitwise addition. The last qubit is flipped if and only if first two qubits are both $|1\rangle$.

satisfying certain symmetry among the constituent qubits, analogous to the parity check condition in classical linear codes.

The fact that the qubit is protected by QECC does not guarantee that one can perform FTQC. When the information is decoded from the QECC, it is no longer protected and becomes vulnerable to error. FTQC thus requires that the entire computation is performed on encoded qubits for protection against errors. For a given physical error rate p , it also requires that all operations on the logical qubits keep the errors within the range correctable by the QECC to first order in p . Finally, the error introduced by qubit measurement process should have probabilities of order p^2 [8]. We will refer to these as FT conditions.

Stabilizer codes are an important class of QECCs where the symmetry of the codewords is described by a set of Pauli operators called *stabilizers* [9]. Valid codewords of a stabilizer code are eigenstates of every stabilizer of the code with eigenvalue of $+1$. We focus as an example on a stabilizer code called the Steane code [10], where $m = 7$ physical qubits are used to encode $k = 1$ bit of information. This code has the capability of correcting one bit-flip error and one phase-flip error on any of the constituent qubits of an encoded state. Furthermore, it has the property that all Clifford group gates can be performed *transversally*, defined as applying the gate in a bit-wise fashion. Since the error from one constituent qubit does not propagate to another qubit in the same code, transversal gates automatically satisfy the FT requirement. The procedures for ensuring FT in stabilizer measurement and non-Clifford group gate necessary for universal QC is well established for the Steane code [11].

Pauli operators (gates) span the operator space of a single qubit, and consist of four operators. Pauli operators on multiple qubits are defined as tensor product of Pauli operators acting on individual qubits. A set of Pauli operators form a group called Pauli Group, denoted by \mathcal{P} . Another important group is called the Clifford group \mathcal{C} consists of operators (gates) that satisfy

$$\mathcal{C} \equiv \{U | \forall P \in \mathcal{P}, UPU^\dagger \in \mathcal{P}\}. \quad (2)$$

\mathcal{C} contains Pauli operators, along with Hadamard gate, CNOT, and phase gate, S.

The Gottesman-Knill theorem shows that it is possible to efficiently simulate a quantum circuit containing only Clifford group gates on a classical computer. In order to achieve the true speed up of a universal QC, one must be able to use at least one non Clifford group gates, such as the $T = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix}$ gate or the Toffoli gate (three qubit control-control-NOT gate, where the state of the target bit is flipped if and only if both control bits are $|1\rangle$) shown in Fig. 1.

2.1. Fidelity as a Performance Metric

Our performance simulator quantifies the correctness of a circuit by recording the fidelity of each qubit during computation. The fidelity between a pure state $|\psi\rangle$ and a mixed state with density matrix $\hat{\rho}$ is defined as $F(\hat{\rho}, |\psi\rangle) = \text{tr} \sqrt{\langle\psi|\hat{\rho}|\psi\rangle}$ [8]. For example, if the initial quantum state is $|\psi\rangle$, then if it is subject to a bit flip error with probability p , the final state density matrix ρ and the corresponding fidelity are given by

$$\rho = (1-p)|\psi\rangle\langle\psi| + pX|\psi\rangle\langle\psi|X \quad (3)$$

$$F = \sqrt{(1-p) + p(\langle\psi|X|\psi\rangle\langle\psi|X|\psi\rangle)}. \quad (4)$$

Fidelity is also defined for a gate operation. If one obtained a density operator ρ_U after applying a gate U on an initial state $|\psi\rangle$ instead of the ideal result $U|\psi\rangle$, the fidelity of the gate operation U is defined as $F(U) \equiv \min_{|\psi\rangle} F(U|\psi\rangle, \rho_U)$, where minimization is taken for all possible initial state $|\psi\rangle$. For the example of the bit flip channel above, $F = \sqrt{1-p} \simeq 1-p/2$.

3. QUIPSIM Overview

The main purpose of our QUantum Information Processor SIMulator (QUIPSIM) is to analyze the interaction of qubits in a quantum algorithm implementing FT procedures. We establish performance metrics for fault tolerant quantum circuits based on the number of ancilla qubits necessary (representing resource overhead), time steps (representing execution time), and overall fidelity of the qubits at the end of computation. Fidelity is a metric chosen to represent both the quality of the qubit and the accuracy of the quantum gate operations (see Sec. 2.1). QUIPSIM keeps track of the local error rates of qubits throughout the QC process as a way to determine error properties of logical qubits, perform resource counting, and optimize FT protocols. The ultimate goal is to bridge the gap between theoretical proofs of FT and realistic error probabilities that must be reached for an experimental realization of a FT quantum processor using a stabilizer code.

The execution complexity of a given logic gate is dictated by the QECC chosen to achieve FT. In the Steane code, gates in the Clifford group are straightforward to perform as they can be implemented transversally. FT implementation of gates outside the Clifford group is highly complex, with increasing levels of complexity for gates with smaller rotational angles or increasing number of control bits [11]. Fortunately, the *Solovay-Kitaev theorem* proves that an arbitrary single qubit gate can be efficiently approximated to arbitrary accuracy by employing just one gate outside Clifford group, typically chosen to be the T gate, to achieve universality [8]. Our non-Clifford group gate of choice is the Toffoli gate (see Fig. 1), as it is very effective in implementing arithmetic logic, such as modular exponentiation, that underlies Shor algorithm. QUIPSIM simulates a universal set of FT quantum gates made of Pauli operators, CNOT, Hadamard, and the Toffoli gate, as well as state preparation, error correction and qubit measurement. This covers all necessary operations to execute a universal FT quantum algorithm using Steane code.

Given a quantum circuit at a logical level, QUIPSIM first breaks each FT logic operation into a set of operations at the constituent level. This provides a much larger and more complex quantum circuit at one concatenation level lower than the original circuit. It performs this circuit expansion iteratively until the circuit reaches the physical level, comprised of qubit initialization, elementary logic gates, and qubit measurement operations. The error probability (or fidelity, see Sec. 2.1) and execution time of each physical operation is provided

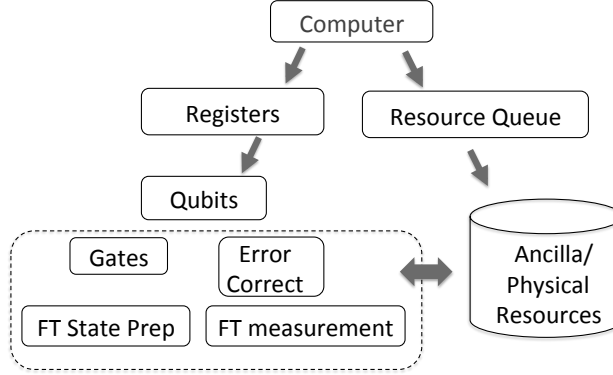


Fig. 2. Code organization for the QUIPSIM program. The computer object is made up of registers which store qubits and a globally shared resource queue of available ancilla and physical resources. Qubit operations, including gates, error correction, state preparation, and measurement, are executed under the constraints given by the availability of ancilla and physical operational resources.

as an input to the simulator, from which the error probability and execution time of a larger block of circuit (*e.g.*, logic gate at a higher level of concatenation) is estimated. Lastly, the effect of decoherence in our system is taken into account by multiplying the fidelity by a factor of $e^{-\lambda\Delta t}$ for a wait time Δt in between gates, with a decoherence rate of λ .

The code organization of the QUIPSIM program is shown in Fig. 2. It implements a hierarchical organization from the computer down to the individual qubits in the registers. The overall quantum computer class contains a sub class for quantum registers to store the qubits and a resource queue to hold the available ancilla and physical resources to run parallel operations. Under the quantum register class is the qubit class which stores the qubits inside the actual registers and performs all the logical and physical level gate operations.

4. Implementation of Quantum Operations

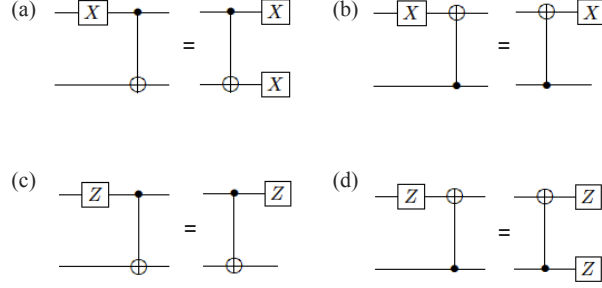
This section summarizes the detailed procedures adopted to break a logical gate operations down into the constituent operations at a lower level of concatenation. We also describe how we account for the fidelity of the resulting qubits in each of these circuit blocks.

4.1. Single Qubit Gates

The single qubit gates chosen in our approach for universal quantum computation include the Pauli operators and the Hadamard operator. Since all of these operators can be implemented transversally, each gate is replaced by the necessary bit-wise single qubit operation at the constituent level. The fidelity of each output constituent qubit is simply the product of the corresponding input constituent qubit and the fidelity of the bit-wise single qubit operation applied to the qubit.

4.2. CNOT Gates and Error Propagation

CNOT gate is also transversal in Steane code, so the logical CNOT gate is replaced by bit-wise CNOT operations at the constituent level. Figure 3 shows the circuit identities that

Fig. 3. Propagation of X and Z errors through the CNOT gates.

describe the processes of "error propagation" through CNOT gate. Figure 3a (3d) shows that X (Z) error on the control (target) qubit propagates to the target (control) qubit, while Figure 3b (3c) shows that X (Z) error on the target (control) qubit is contained in the target (control) qubit and does not propagate to the control (target) qubit. In most cases, we make a simplifying assumption that the application of a CNOT gate operation degrades the fidelity of the output qubits by multiplying the fidelities of both the input qubits together and then multiplying that by the fidelity of the gate operation itself. This provides the worst-case estimate on the fidelity of the output qubit, as some errors do not propagate through the CNOT gate to degrade the fidelity of the resulting qubits.

4.3. Cat State Preparation

One common procedure which must be implemented in stabilizer-based FT quantum computation is preparation of maximally entangled states called cat states, used primarily for FT measurement. The typical preparation scheme for a four qubit cat state is shown in Fig. 4. The first four qubits form a cat state after the Hadamard and first three CNOT gates are applied. The probability that the cat state generated contains errors is proportional to the error rate of the single gate error rate p , and is not good enough for use in FT measurement. A fifth verification qubit is introduced to check the parity of the first and the last qubit in the cat state, by the last two CNOT gates and the measurement. If the parity is odd, the cat state is discarded and the cat state preparation is repeated. If the parity is even, the generated cat state has an error probability of $O(p^2)$, and is ready for use.

To verify the error probability of the cat state prepared in this way, we trace the fault paths of possible X and Z errors through the circuit in Fig. 4. If we let p_s and p_c be the probability of having an error during physical state preparation and CNOT gate, respectively, we note that any single X error in the process will propagate to the verification qubit and be detected. The final fidelity of an l -qubit cat state is derived by counting the number of ways in which the verification qubit fails to detect one or more errors remaining in the prepared cat state. Fig. 3 shows an example of such an occasion, where the physical state preparation error in the first qubit is cancelled at the verification qubit by an error in the first CNOT operation (modeled as X error in the target qubit following the CNOT gate), resulting in two qubits flipped in the cat state. The fidelity for each of the l constituent qubits in the resulting cat

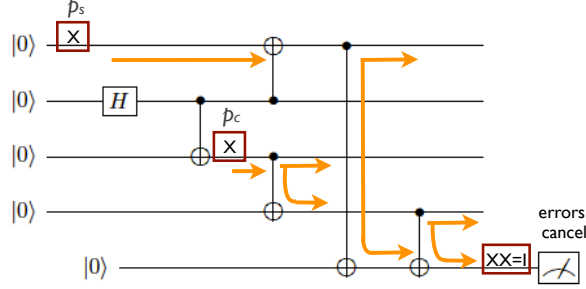


Fig. 4. Preparation of a cat state with an example of error propagation through the circuit. The probability of two errors canceling each other, as shown here, is equal to $(l+1)p_s p_c$ for an l -qubit cat state.

state is given by $f_{cat} \simeq 1 - \frac{1}{2}[(l-1)p_s^2 + (l+1)p_s p_c + (l+1)p_c^2] + O(p^3) \sim 1 - (3l+1)p^2/2$ when $p_s = p_c \equiv p$, satisfying the FT condition (3). We note that Z errors cannot be detected in this scheme, but show in the next section that these cat states can still be used to effectively perform FT stabilizer measurements.

4.4. State Preparation and Logical State Detection

QUIPSIM uses the Steane encoding scheme to prepare the logical $|0\rangle$ states, achieved by measuring the six stabilizers on an arbitrary (mixed) state of seven physical qubits. Each stabilizer measurement uses a 4-qubit cat state, as shown in Fig. 5 [8]. We choose the approach that uses six 4-qubit cat states to measure the stabilizers simultaneously rather than recycle a single cat state, to greatly reduce the execution time for the state preparation. We note that an efficient protocol exists that uses decoding circuits to overcome the slow measurement processes that forced us to parallelize the stabilizer measurements [12]. After all the CNOT gates are executed, we measure the first qubit in each of the six cat states to determine the eigenvalue of the stabilizer. If the eigenvalue -1, we apply the logical X operation to adequate constituent qubit which flips the state back to +1 eigenstate [9].

If the cat state used for the measurement is prepared using procedure shown in Fig. 4, the stabilizer measurement is FT against a single X error propagating into the data qubits. Any Z error in either the cat state preparation or in the stabilizer measurement process does not propagate into the data qubits (due to propagation shown in Fig. 3c), but will result in flipping the first qubit in the cat state after the Hadamard gate leading to an error in the stabilizer readout. Error in the measurement process itself leads to a similar result in the stabilizer readout with error rate $O(p)$. Since correction operations on the data qubits are performed based on the result of the stabilizer measurement, this will lead to an error probability in the data qubit of $O(p)$, violating the FT condition. To overcome this, QUIPSIM performs the measurement of each stabilizer generator three times and uses the majority vote on the (classical) readout results to decide if the correction operation should be applied. This reduces the error probability of the measurement (originating either from Z errors in cat states or the measurement process itself) to $p_m \approx 3p^2$. The final fidelity of each physical qubit in the

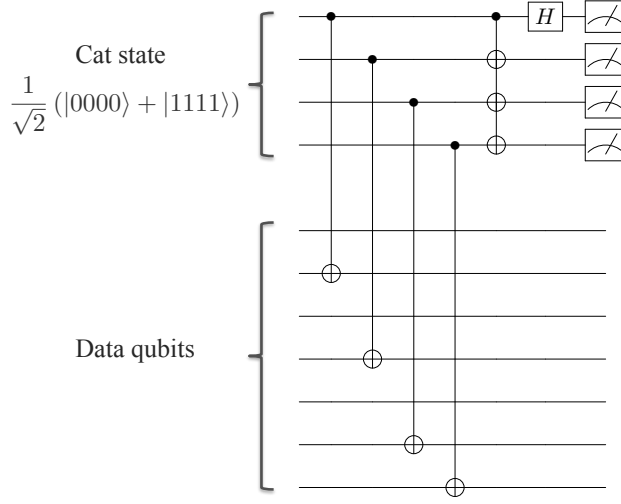


Fig. 5. FT measurement of the $g_2 = IXXIIXX$ generator of the Steane code using a 4- qubit cat state. Stabilizer measurements such as this are used in both state preparation and in error correction.

logical encoded state being prepared is then equal to this error probability of measurement multiplied by the error probability of the potential correction gate applied, giving a state preparation error rate of $p_{s.p.} = p(1 - 3p^2)$ and fidelity $f_{s.p.} = 1 - p_{s.p.}/2$. This ensures that the probability of having two errors in the codeword is $O(p^2)$, satisfying the FT condition.

After all six stabilizer measurements are performed, we must perform a logical Z measurement to make sure the logical qubit is prepared as $|0\rangle$. This process is done by preparing a 3-qubit cat state and measuring the parity of the first three qubits: this parity is measured three times to reduce the measurement-induced error to $O(p^2)$. If the Z measurement indicates a result -1, then the state $|1\rangle$ is prepared: we apply the bit-wise X operation on the first three constituent qubits to flip the state back to $|0\rangle$. The resulting fidelity of the first three qubits in a logical qubit state preparation therefore get an additional factor of $(1 - p)$ from the X operation. This procedure of logical Z measurement also serves as the qubit state detection at the logical level.

4.5. Error Correction

Part of the state preparation process, namely measuring the six stabilizers and applying associated correction factors, is a sufficient error correction procedure. QUIPSIM uses the state preparation process of Sec. 4.4 less the logical Z measurement as the error correction procedure, which ensures that the qubit state is projected back into the code space without measuring the qubit itself. The fidelity of the qubits after error correction is calculated by adding the probability of two situations where (1) data qubits do have an error, and the syndrome measurement does not detect an error, and (2) data qubits had no error but the syndrome detection detects one, and a correction procedure was applied. This estimation accurately reduces the remaining error probability after error correction is applied to $O(p^2)$. All other situations lead to error probabilities of $\sim O(p^3)$, and can be safely ignored in the limit

of low physical error probabilities. QUIPSIM implements this by calculating the probability that the stabilizer measurements during error correction have an error. Since measurements are repeated three times, the final fidelity is reduced to $f \approx 1 - 9p_{meas}^2$. While this is just an approximate expression, the implementation by the simulator computes a more accurate value by finding the measurement failure rate p_{meas} for each of the three measurements, and combining them to get the probability that the error correction mistakenly finds an error when none is present or does not detect an error when one is present.

4.6. Fault tolerant Toffoli gate

Universal quantum computation requires at least one gate outside the Clifford group; here we choose the Toffoli gate. Unlike the Clifford group, a fault tolerant version of the Toffoli gate cannot be implemented transversally. A fault tolerant version of the Toffoli requires a 7-qubit cat state and three ancilla qubits prepared in a known state $|000\rangle$, as shown in Fig. 6a [11]. The top qubit is the 7-qubit cat state, and the CNOT, Hadamard, Z and measurement are performed using the FT procedures described above. The Toffoli gate shown here is actually a bit-wise Toffoli performed on each constituent qubits. Bit-wise Toffoli itself is not a FT implementation of Toffoli gate, but in this circuit we are measuring the eigenvalue of CNOT operation performed between the third and the fourth qubit, by performing bit-wise controlled-CNOT with the 7-qubit cat state. The bit-wise Toffoli can be further broken down into lower-level circuit if it is still encoded. We have to assume that our physical hardware can perform a Toffoli gate at the physical level, where experimental demonstrations exist [15]. The resulting three-qubit state replaces the first two Hadamard gates and the Toffoli gate in the circuit shown in Fig. 6b, which shows a FT implementation of the Toffoli gate. All remaining operations consist of CNOTs, controlled- Z gates (which can be implemented with one CNOT and two Hadamard gates), qubit measurements and single qubit gates, which can be simulated fault tolerantly by QUIPSIM.

Figure Fig. 6c shows the error probability of the FT Toffoli gate circuit with one level of encoding as a function of the physical error probability. It shows that below the physical error probabilities of $p \sim 4 \times 10^{-5}$, the encoded gate shows much better error probabilities than the uncoded one. This was taken in the no memory error limit where the decoherence rate is much longer than the gate time. Since the FT Toffoli gate has the highest error rate of all the gates in the universal set described here, this error value can be thought of as a threshold error rate since the FT implementation of any gate decreases the overall error rate when the physical rate is lower than this value.

5. Quantum adder circuits

The most celebrated application of quantum computation is Shor's algorithm, which involves modular exponentiation (ME) and quantum Fourier transform (QFT) [1]. An n -bit ME is a classical arithmetic circuit, but requires $O(n^3)$ operations compared to the $O(n^2)$ operations for QFT and dominates the computation time. The algorithmic optimization of ME circuit has been considered, and the performance is dictated by the ability to implement efficient quantum adders [16]. Several authors have proposed optimized versions of various known adder circuits to fit into quantum domain [17, 18, 19, 20], reducing the number of resource-intensive Toffoli gates and depth of the circuit. As an example application

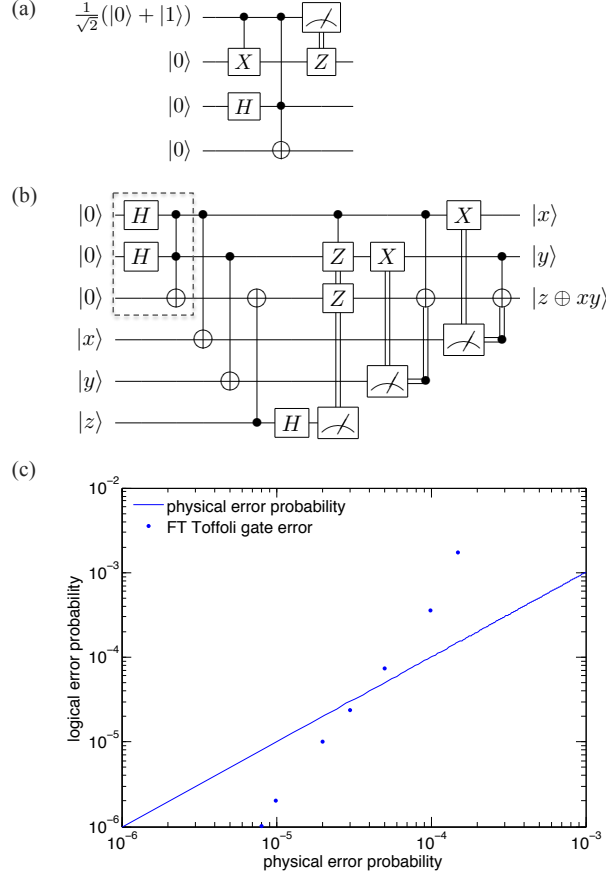


Fig. 6. A FT Toffoli gate. The Toffoli and Hadamard gates in the box in circuit in (b) can be replaced with the operations shown in (a). (c) Error probability of the FT Toffoli gate as a function of the physical error probability.

of QUIPSIM, we compare the execution time, resource requirements and error properties of two adders: quantum ripple carry adder (QRCA) and the quantum carry look-ahead adder (QCLA). Optimized QRCA features $2n + O(1)$ Toffoli gates and has depth $2n + O(1)$ for n -bit addition [17]. QCLA features $5n - O(\log n)$ Toffoli gates and has $O(2 \log n)$ depth [18].

6. Performance Evaluation

In this section we show some performance simulation examples demonstrating the capabilities of QUIPSIM. Although QUIPSIM is capable of independently setting the operation times and error probabilities for each physical gate type, we made the simplifying assumption to normalize every physical gate time to 1 unit with an associated error probability of p in the examples shown here to determine the more generic behavior of the circuits. We also simplify the memory error by setting the decoherence rate to $\lambda = 0$ in order to isolate the effects of gate error in a system with no memory errors. Physical measurement times are

assumed to be 200 times the gate time, to reflect the realities of many experimental situations where state readout is significantly slower than qubit gates. One important assumption we made was regarding the connectivity of the qubits in the system. In any experimental setup, detailed physical implementation of a two qubit gate between a pair of qubits in the system necessarily depends on their relative location, providing additional constraints in the performance measures. One should make assumptions about how to realize two qubit gates between distant pairs of qubits. In our model, we make the simplifying assumption that a two qubit gate can be applied between any pair of qubits in the system, i.e. full connectivity.

The performance of QRCA and QCLA circuits is evaluated with and without the use of FT encoding. Fig. 7a shows the execution time of the two circuits measured in units of physical gate times. While the fault tolerant encoding of the circuits gives a substantial increase in the execution time (around three orders of magnitude), it does not change the nature of the circuit depth and hence the QCLA circuit still scales logarithmically in the adder size with approximately the same scale factor as the unencoded adder. Fig. 7b shows the average error probability of the output qubits for the FT QCLA circuit as a function of the adder size n . At error probabilities below threshold ($p_{th} \sim 4 \times 10^{-5}$), the final error rate of the circuit is much lower than the physical error rate, as FT behavior would suggest. The transition above threshold is seen in cases where sharp decline in final fidelities is seen with increasing adder size.

7. Conclusion

QUIPSIM is a software tool capable of simulating the performance of a complete set of FTQC protocols in the presence of faulty gates and low error rates. QUIPSIM tracks the fidelity of each qubit over the course of the circuit using two basic approaches: (1) in state preparation and error correction blocks, it explicitly computes the number of fault paths that lead to erroneous outcome to estimate the fidelity, and (2) in Clifford group gates, it performs the error estimation by keeping track of the worst-case scenario of how errors propagate through these gates. This approach provides a conservative estimate of the error behavior of the overall circuit, but is very effective in low-error systems where other approaches like Monte-Carlo simulations take a very long time to converge. We used QUIPSIM to test the performance of two types of quantum adder circuits of various sizes, employing Steane code for the FT procedure.

QUIPSIM has the flexibility to add more realistic hardware models by defining the execution time and error probabilities for each gate type. We can further improve the FT protocol by utilizing more efficient stabilizer measurement processes. QUIPSIM can be used to study the relationship between physical gate parameters and the FT execution of a quantum circuit to optimize FT procedures. One limitation of the current version is that the qubits are assumed to be fully connected. An important extension of this work is to evaluate the performance of quantum circuits on physical hardware with added constraints like the connectivity of qubits and availability of resources, leading to the ability to optimize the hardware architecture of quantum processors based on experimental constraints.

Acknowledgements

We would like to thank Byung-Soo Choi and Rachel Noek for helpful discussions. This work

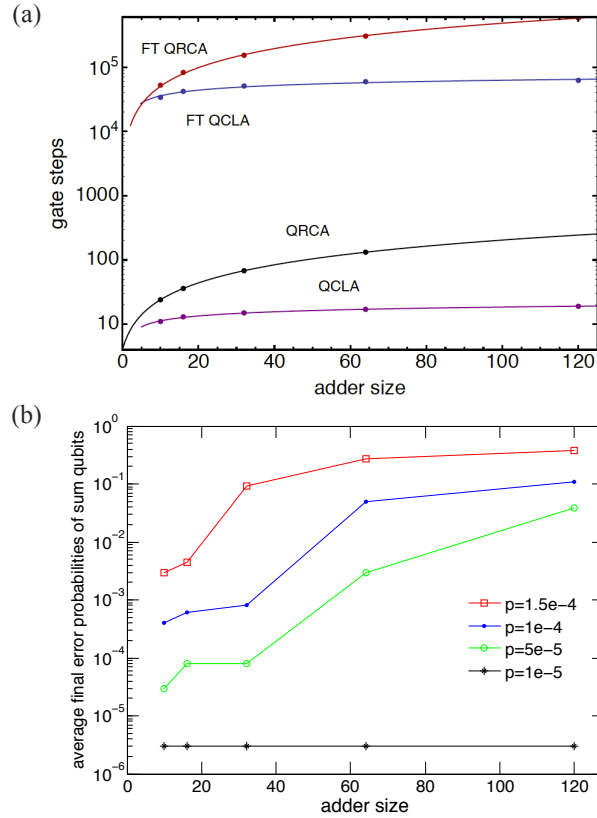


Fig. 7. Simulation of FT QCLA and ripple carry adder circuit. (a) Execution time of the QRCA and QCLA circuits with and without FT encoding, in units of physical gate times. (b) Average error probabilities of the output qubits for the FT QCLA circuit. The final error probability is independent of adder size when the gate error rate is $p = 1 \cdot 10^{-5}$ since this is effectively below threshold.

was supported by the NSF and IARPA

References

1. P. W. Shor, "Algorithms for quantum computation: discrete logarithms and factoring," in *Proc. 35th Annual Symposium on Foundations of Computer Science*, pp 124, 1994.
2. L. K. Grover, "Quantum mechanics helps in searching for a needle in a haystack," *Phys. Rev. Lett.* 79, pp 325, 1997.
3. T. S. Metodi, D. D. Thaker, A. W. Cross, F. T. Chong, and I. L. Chuang, "Scheduling physical operations in a quantum information processor," *Proc. of SPIE* 6244, 62440T, 2006.
4. D. Aharonov and M. Ben-Or, "Fault-tolerant quantum computation with constant error," *Proc. 29th Annual ACM Symposium on Theory of Computing*, pp 176, 1997.
5. A.W. Cross, D.P. DiVincenzo, B.M. Terhal, "A comparative code study for quantum fault tolerance." *Journal of Quantum Inf. and Comp.* Volume 9 Issue 7, 541-572 (2009).
6. P. Aliferis, D. Gottesman, and J. Preskill, "Quantum Accuracy Threshold for Concatenated Distance 3 Quantum Codes." *Journal of Quantum Inf. and Comp.* Volume 6 Issue 2, 97-165 (2006)
7. E. Knill and R. Laflamme, "Theory of quantum error-correcting codes," *Phys. Rev. A* 55, 900,

- 1997.
8. M. Nielsen and I. Chuang, *Quantum Computation and Quantum Information*, Cambridge University Press, 2000.
9. D. Gottesman, "A theory of fault-tolerant quantum computation," *Phys. Rev. A* 57, pp 127, 1998.
10. A. M. Steane, "Error correcting codes in quantum theory," *Phys. Rev. Lett.* 77, pp 793, 1996.
11. X. Zhou, D. W. Leung, and I. L. Chuang, "Methodology for quantum logic gate construction," *Phys. Rev. A* 62, 052316, 2000.
12. D. P. DiVincenzo and P. Aliferis, "Effective fault-tolerant quantum computation with slow measurements," *Phys. Rev. Lett.* 98, 020501, 2007.
13. A. M. Steane, "Overhead and noise threshold of fault-tolerant quantum error correction," *Phys. Rev. A* 68, 42322, 2003.
14. P. Aliferis, "Level reduction and the quantum threshold theorem," Ph.D. Thesis, California Institute of Technology, 2007.
15. T. Monz, K. Kim, W. Hänsel, M. Riebe, A. S. Villar, P. Schindler, M. Chwalla, M. Heinrich, and R. Blatt, "Realization of the quantum toffoli gate with trapped ions," *Phys. Rev. Lett.* 102, 040501, 2009.
16. R. Van Meter and K. M. Itoh, "Fast quantum modular exponentiation," *Phys. Rev. A* 71, 052320, 2005.
17. A. Cuccaro, T. G. Draper, S. A. Kutin, and D. P. Moulton, "A new quantum ripple-carry addition circuit," <http://arXiv.org/quant-ph/0410184>, 2004.
18. T. G. Draper, S. A. Kutin, E. M. Rains, and K. M. Svore, "A logarithmic-depth quantum carry-lookahead adder," *Quant. Inf. Comput.* 6, pp 351, 2004.
19. T. G. Draper, "Addition on a quantum computer," <http://arXiv.org/quantph/0008033>, 2000.
20. P. Gossett, "Quantum carry-save arithmetic," <http://arXiv.org/quant-ph/9808061>, 1998.